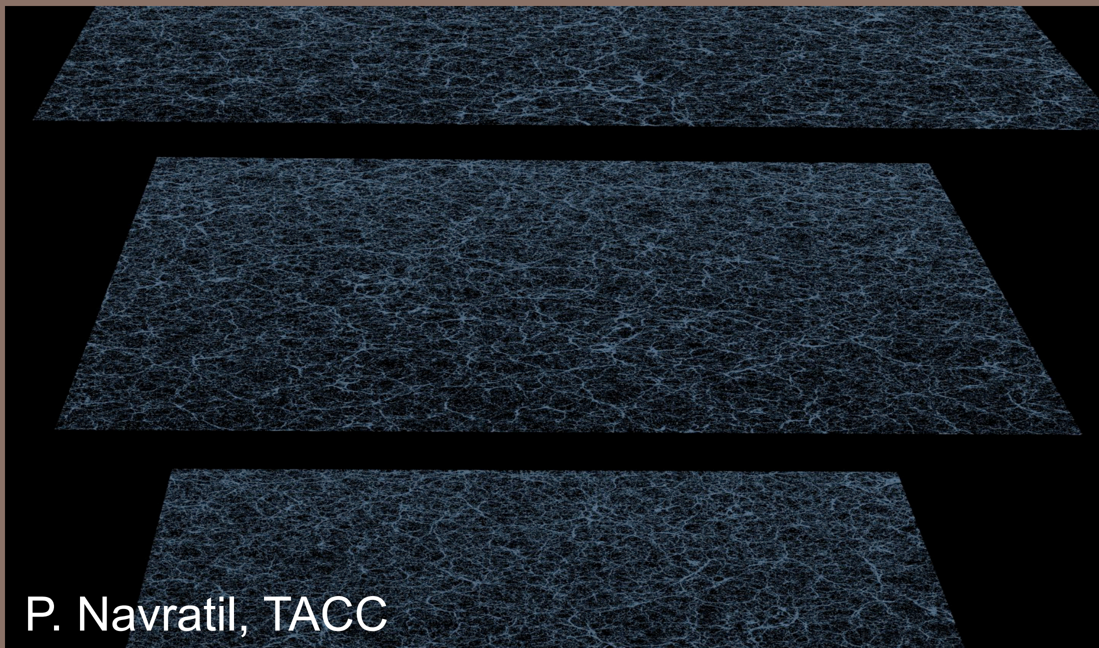
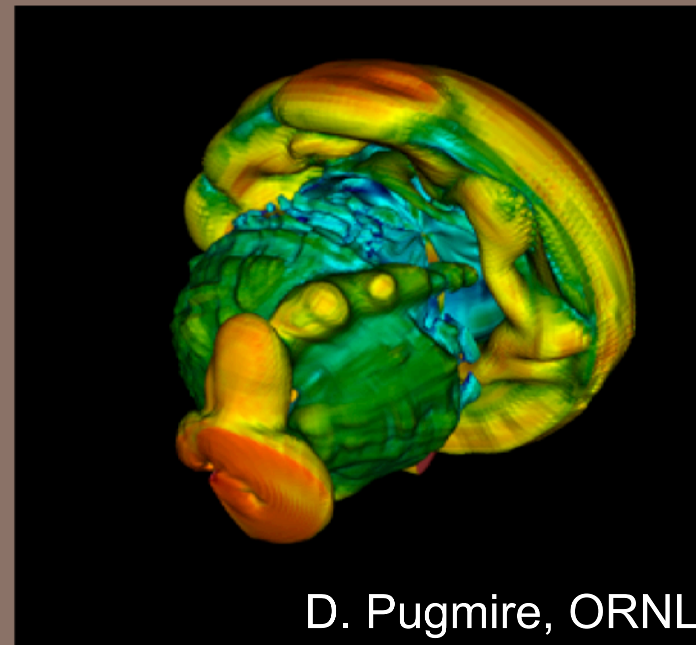




WHY THE RULES ARE CHANGING FOR LARGE DATA VISUALIZATION AND ANALYSIS

11/2/10

Hank Childs
Lawrence Berkeley Lab & UC Davis

Supercomputing 101

- Why simulation?
  - Simulations are sometimes more cost effective than experiments.
  - New model for science has three legs: theory, experiment, and *simulation.*
- What is the “petascale” / “exascale”?
  - 1 FLOP = 1 FLoating point OPeration per second
  - 1 GigaFLOP = 1 billion FLOPs, 1 TeraFLOP = 1000 GigaFLOPs
  - 1 PetaFLOP = 1,000,000 GigaFLOPs, 1 ExaFLOP = billion billion FLOPs
  - PetaFLOPs + petabytes on disk + petabytes of memory → petascale
  - ExaFLOPs + exabytes on disk + petabytes of memory → exascale
- Why petascale / exascale?
  - More compute cycles, more memory, etc, lead for faster and/or more accurate simulations.

Petascale computing is here.

- Existing petascale machines

LANL RoadRunner

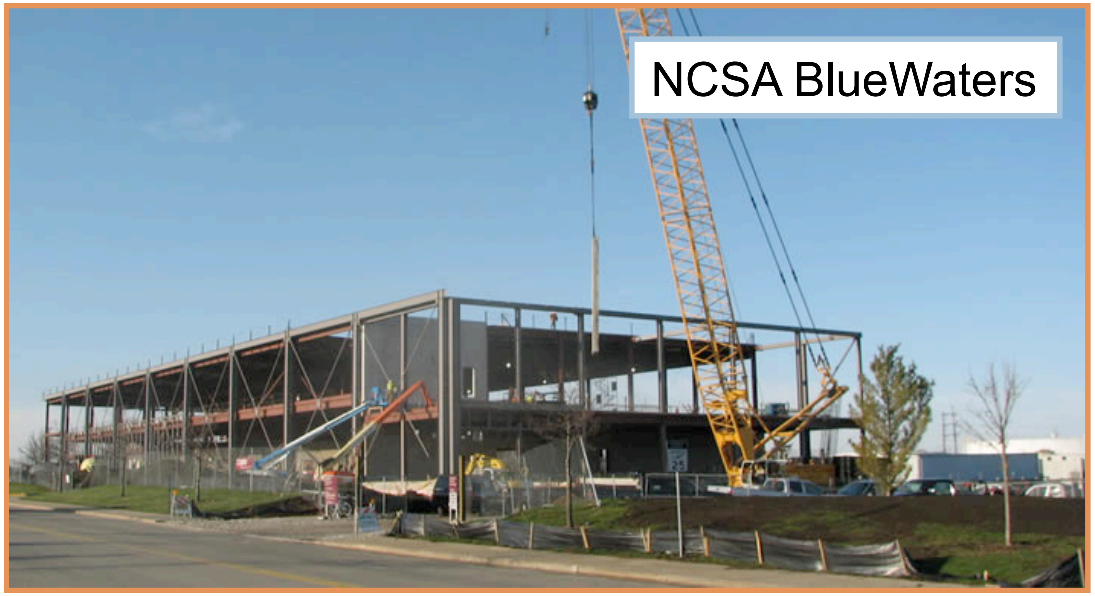
ORNL Jaguar

UTK Kraken

Julich JUGene

Supercomputing is not slowing down.

- Two ~20 PetaFLOP machines will be online in 2011

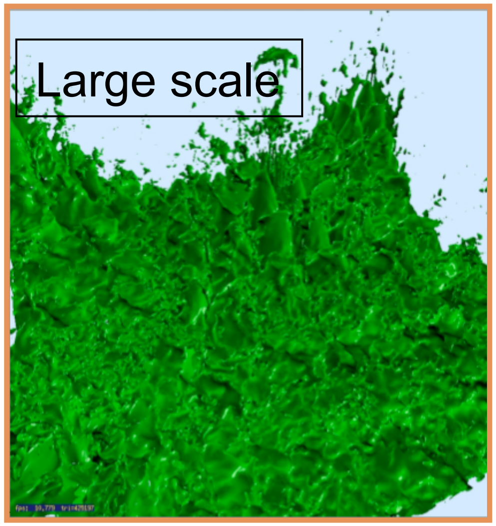


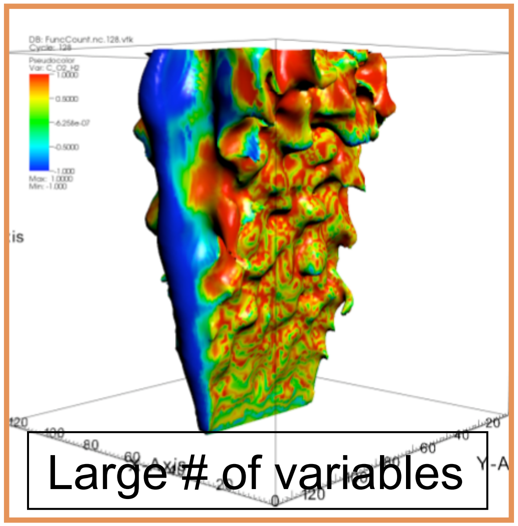
- Q: When does it stop?
- A: Exascale is being actively discussed right now
  - http://www.exascale.org

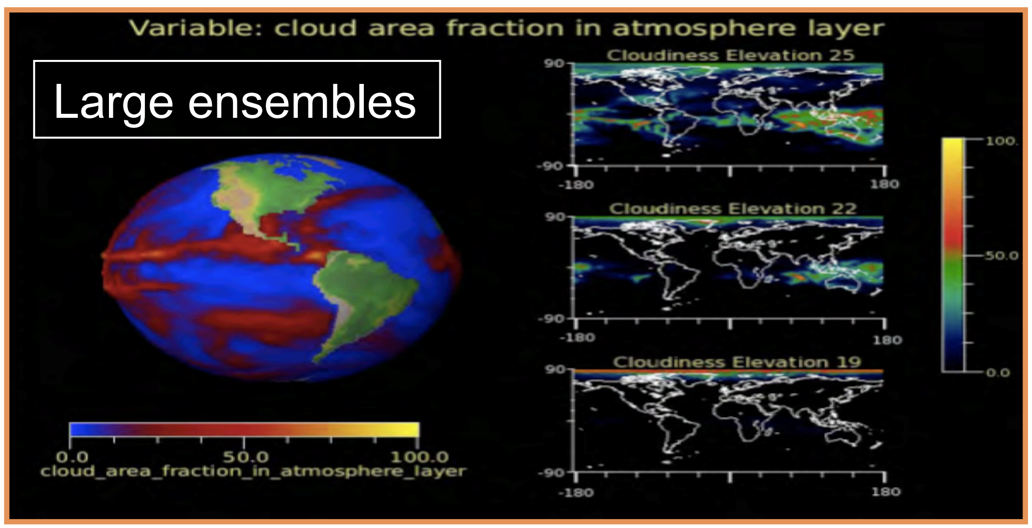
Exascale machine: requirements

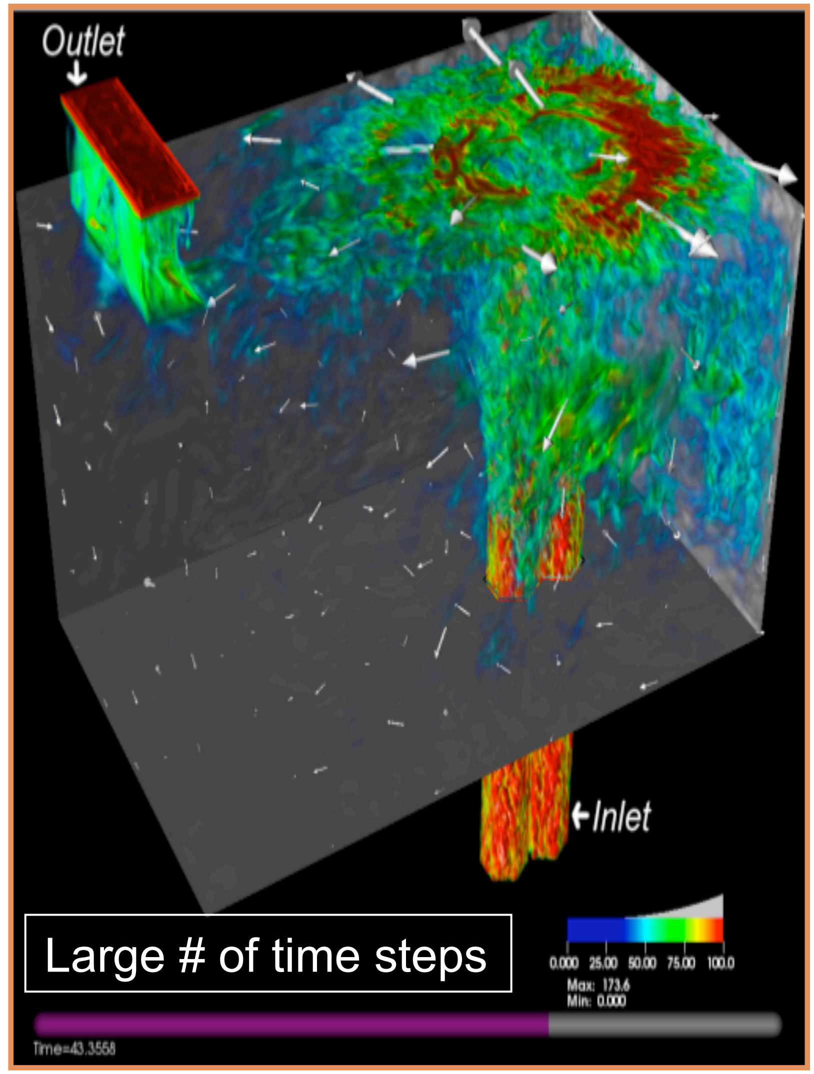
- Timeline: 2018-2021
- Total cost: <$200M
- Total power consumption: < 20MW
- Accelerators a certainty
- FLASH drives to stage data will change I/O patterns (very important for vis!)

How does the petascale/exascale affect visualization?









Why is petascale/exascale visualization going to change the rules?

- Michael Strayer (U.S. DoE Office of Science): *“petascale is not business as usual”*
  - Especially true for visualization and analysis!
- Large scale data creates two incredible challenges: d complexity

Outline

- “business as usual”
  - Supercomputing landscape is changing
  - Solution: we will need “smart” techniques in production environments
- More resolution leads to more and more complexity
  - Will the “business as usual” techniques still suffice?

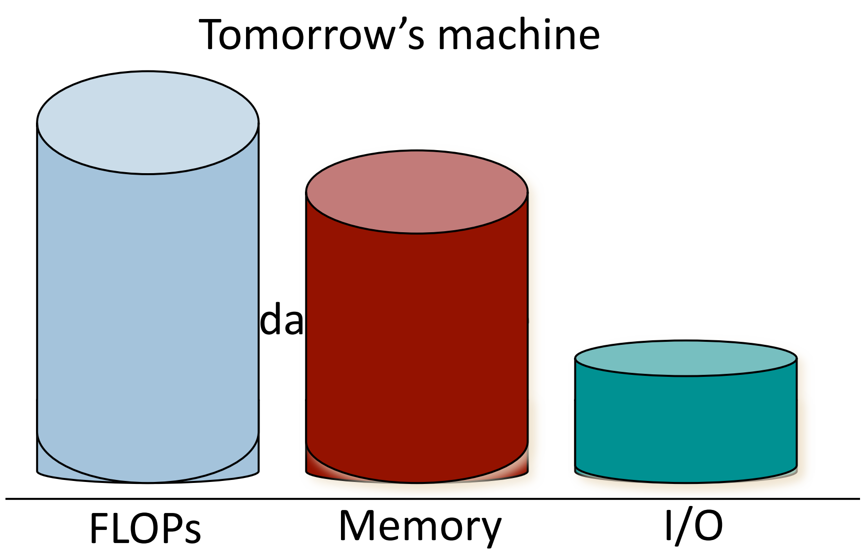
Production visualization tools use "pure parallelism" to process data.

Pure parallelism: pros and cons

- Pros:
  - Easy to implement
- Cons:
  - Requires large amount of primary memory
  - Requires large I/O capabilities
  - → requires big machines

Pure parallelism performance is based on # bytes to process and I/O rates.

- Amount of data to visualize is typically O(total mem)
- Vis is almost always >50% I/O and sometimes 98% I/O



- Two big factors:
  1. how much data you have to read
  2. how fast you can read it
- → Relative I/O (ratio of total memory and I/O) is key

Anedoctal evidence: relative I/O is getting slower.

Time to write memory to disk

Machine name	Main memory	I/O rate	
ASC purple	49.0TB	140GB/s	5.8min
BGL-init	32.0TB	24GB/s	22.2min
BGL-cur	69.0TB	30GB/s	38.3min
Sequoia	??	??	>>40min

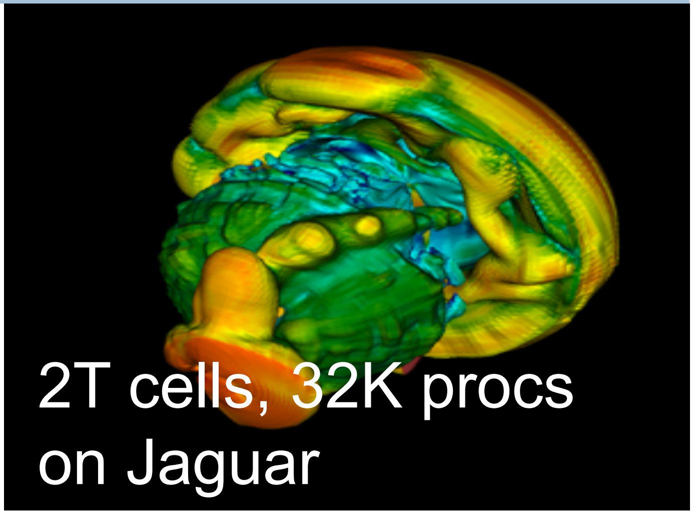
Why is relative I/O getting slower?

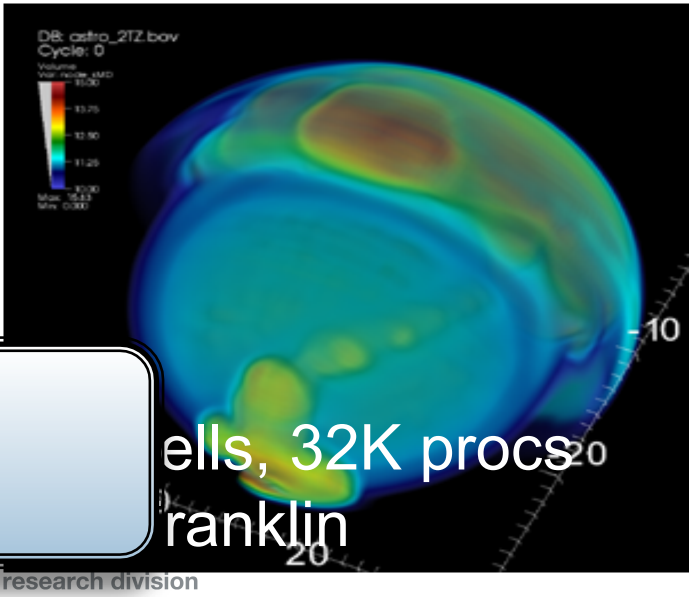
- "I/O doesn't pay the bills"
  - And I/O is becoming a dominant cost in the overall supercomputer procurement.
- Simulation codes aren't as exposed.
  - And will be less exposed with proposed future architectures.

Recent runs of trillion cell data sets provide further evidence that I/O dominates

- Weak scaling study: ~62.5M cells/core

Machine	Type	Problem Size	#cores
Franklin	Cray XT4	1TZ, 2TZ	16K, 32K
Dawn	BG/P	4TZ	64K
JaguarPF	Cray XT5	2TZ	32K
Juno	Linux	1TZ	16K
Purple	AIX	0.5TZ	8K
Ranger	Sun Linux	1TZ	16K



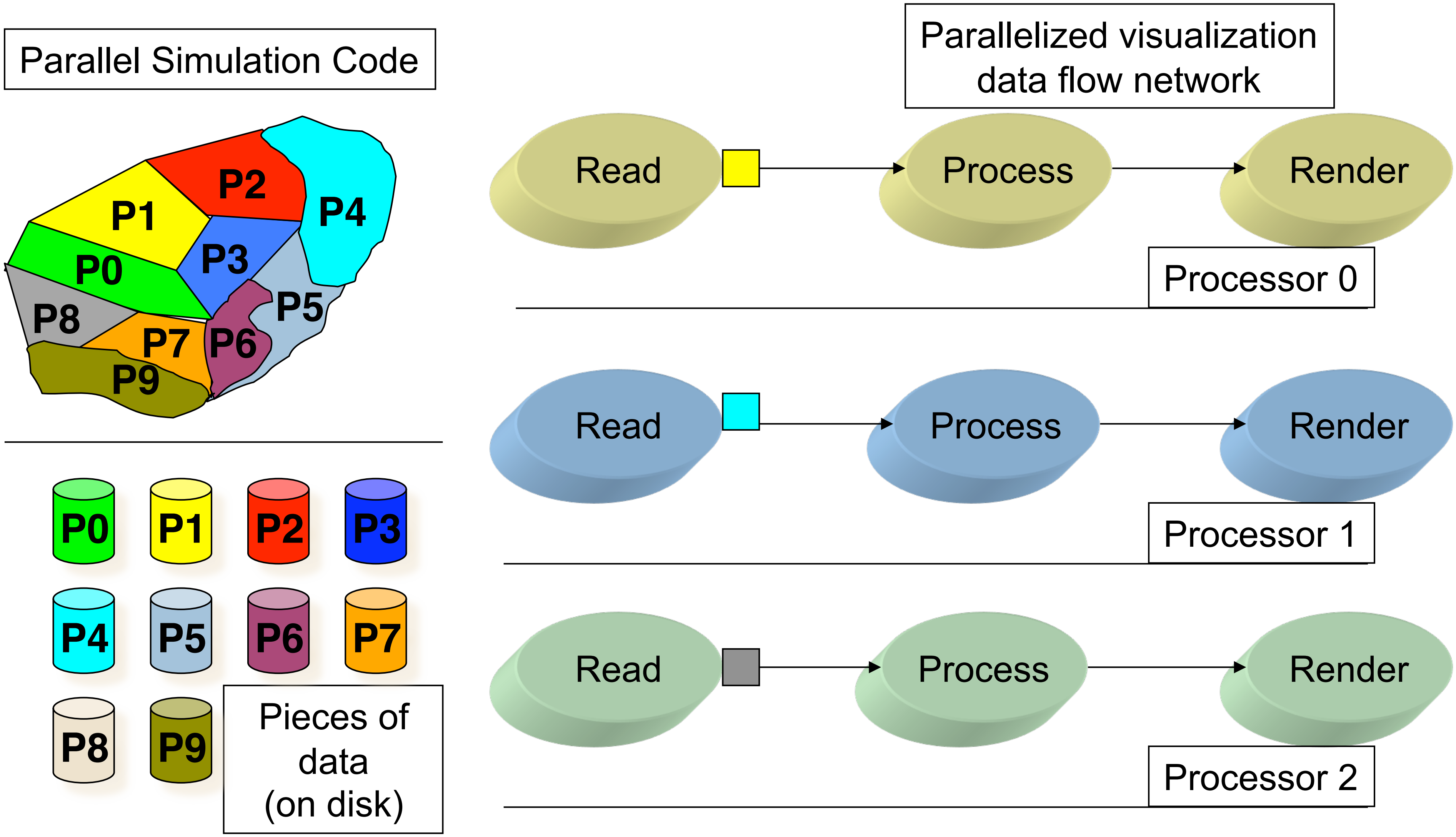


-Approx I/O time: 2-5 minutes
-Approx processing time: 10 seconds

Pure parallelism is not well suited for the petascale.

- Emerging problem:
  - Pure parallelism emphasizes I/O and memory
  - And: pure parallelism is the dominant processing paradigm for production visualization software.
- Solution? … there are "smart techniques" that de-emphasize memory and I/O.
  - Data subsetting
  - Multi-resolution
  - Out of core
  - In situ

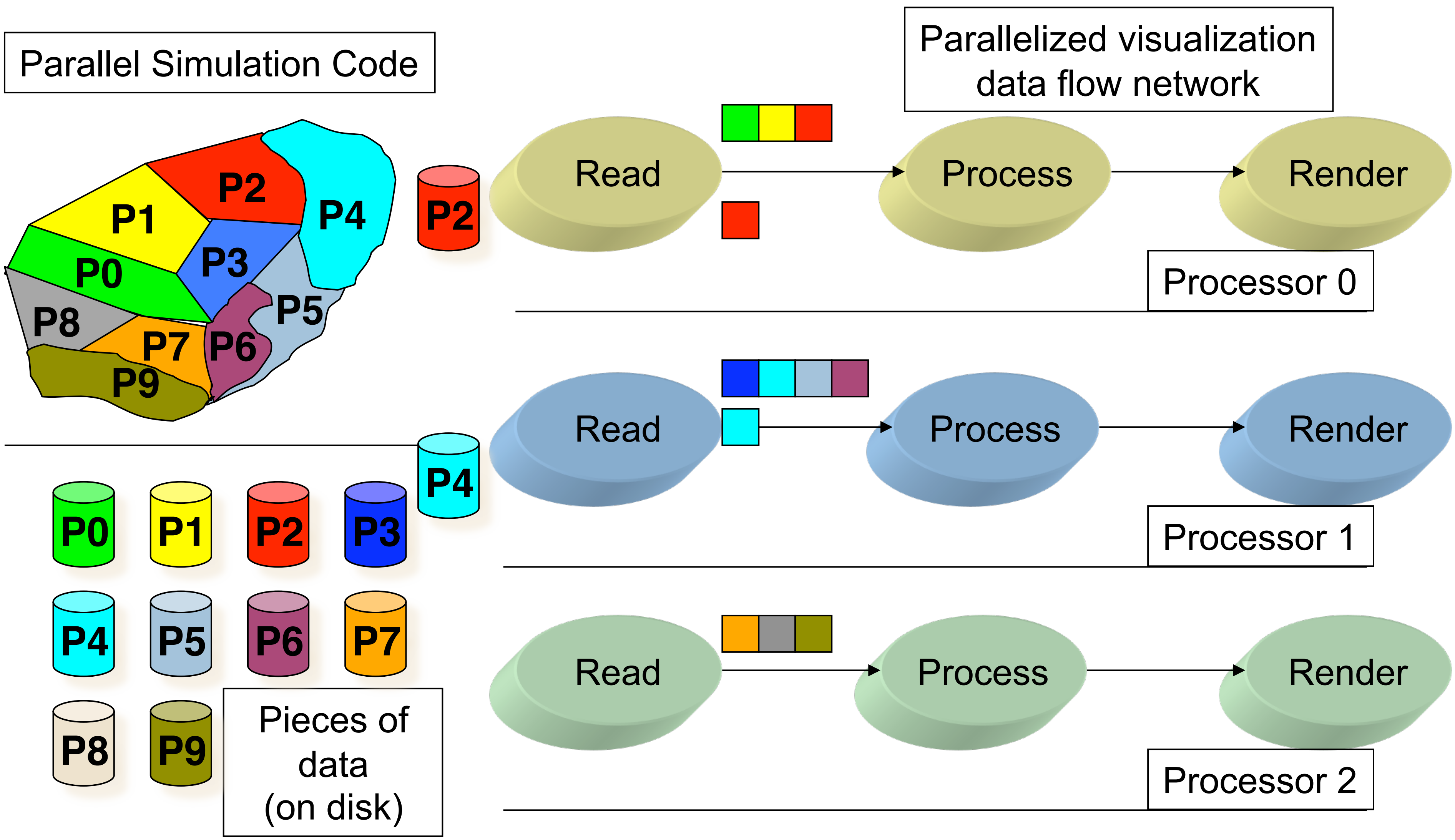
Data subsetting eliminates pieces that don't contribute to the final picture.



Data Subsetting: pros and cons

- Pros:
  - Less data to process (less I/O, less memory)
- Cons:
  - Extent of optimization is data dependent
  - Only applicable to some algorithms

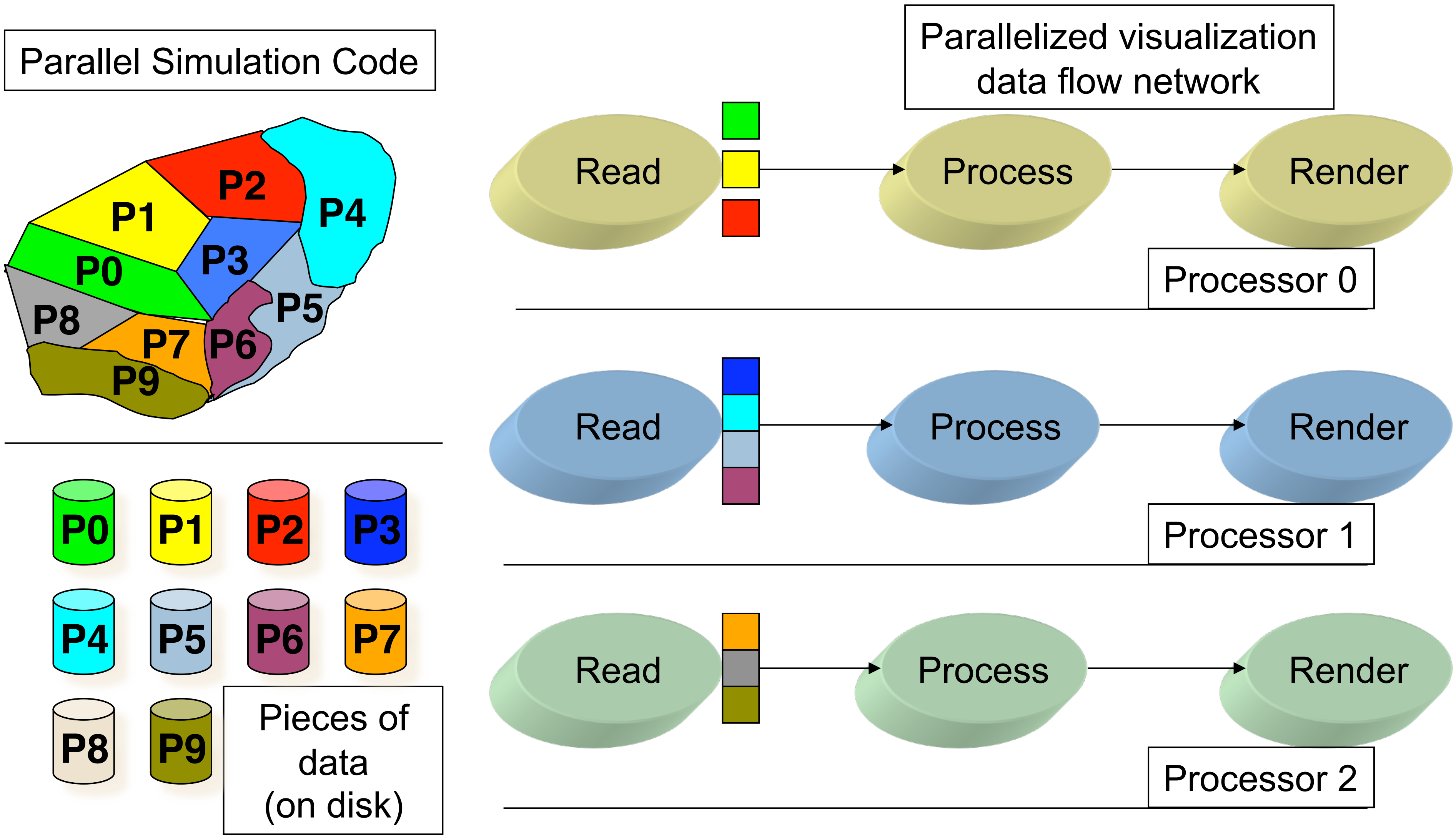
Multi-resolution techniques use coarse representations then refine.



Multi-resolution: pros and cons

- Pros
  - Avoid I/O & memory requirements
- Cons
  - Is it meaningful to process simplified version of the data?

Out-of-core iterates pieces of data through the pipeline one at a time.

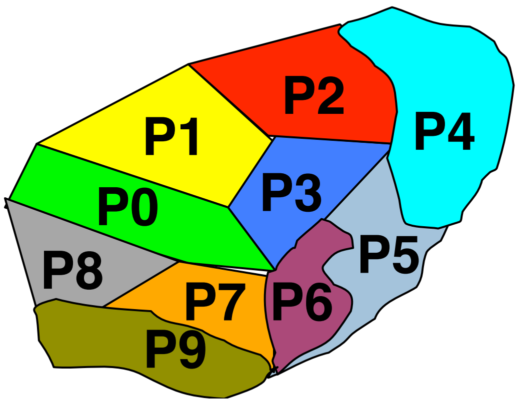


Out-of-core: pros and cons

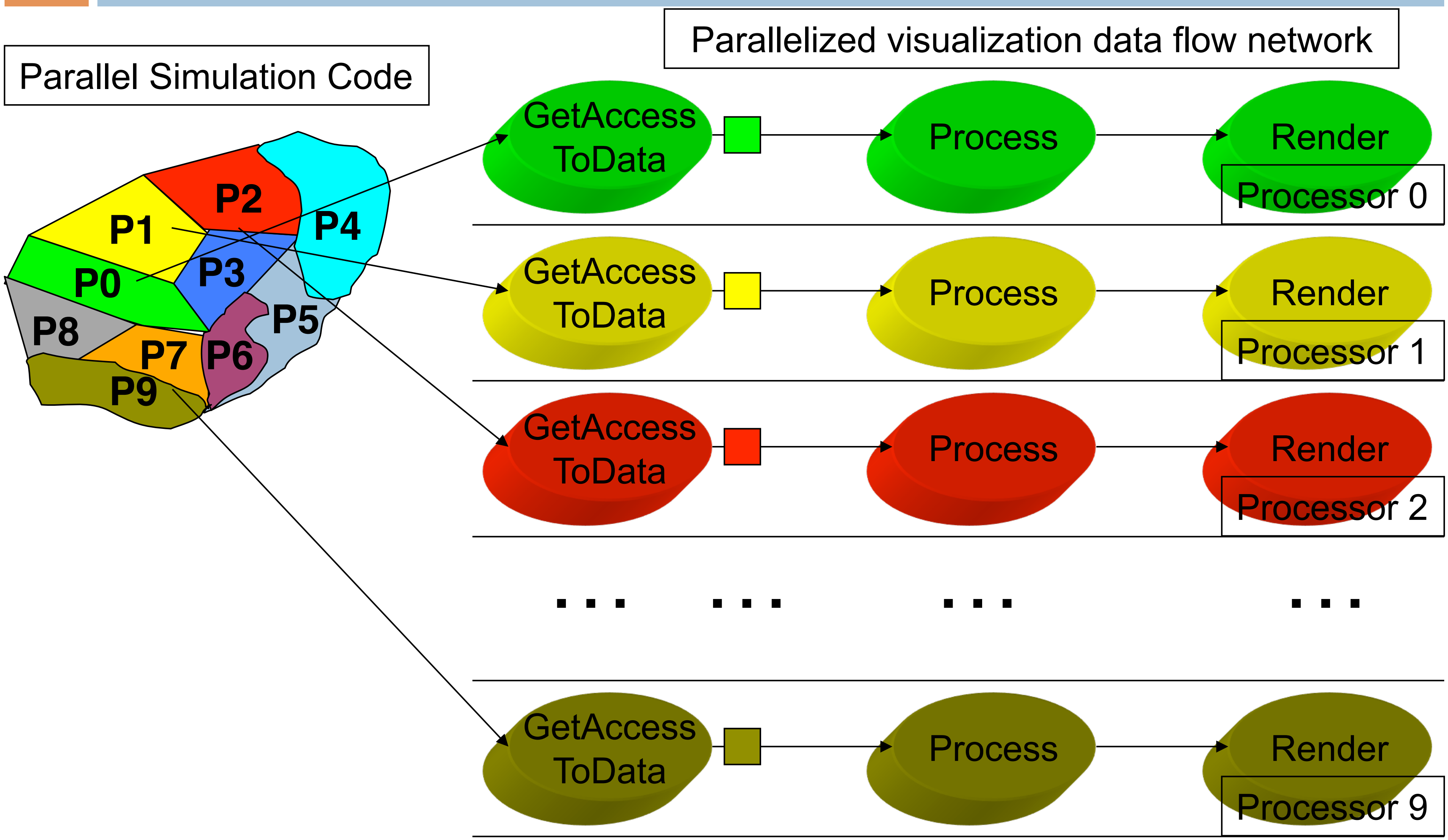
- Pros:
  - Lower requirement for primary memory
  - Doesn't require big machines
- Cons:
  - Still paying large I/O costs
    - (Slow!)

In situ processing does visualization as part of the simulation.

Parallel Simulation Code



In situ processing does visualization as part of the simulation.



In situ: pros and cons

- Pros:
  - No I/O!
  - Lots of compute power available
- Cons:
  - Very memory constrained
  - Many operations not possible
    - Once the simulation has advanced, you cannot go back and analyze it
  - User must know what to look a priori
    - Expensive resource to hold hostage!

Summary of Techniques and Strategies

- Pure parallelism can be used for anything, but it takes a lot of resources
- Smart techniques can only be used situationally
- Strategy #1 (do nothing):
  - Stick with pure parallelism and live with high machine costs & I/O wait times
- Other strategies?
  - Assumption:
    - We can't afford massive dedicated clusters for visualization
    - We can fall back on the super computer, but only rarely

Now we know the tools … what problem are we trying to solve?

- Three primary use cases:
  - Exploration
  - Confirmation
  - Communication

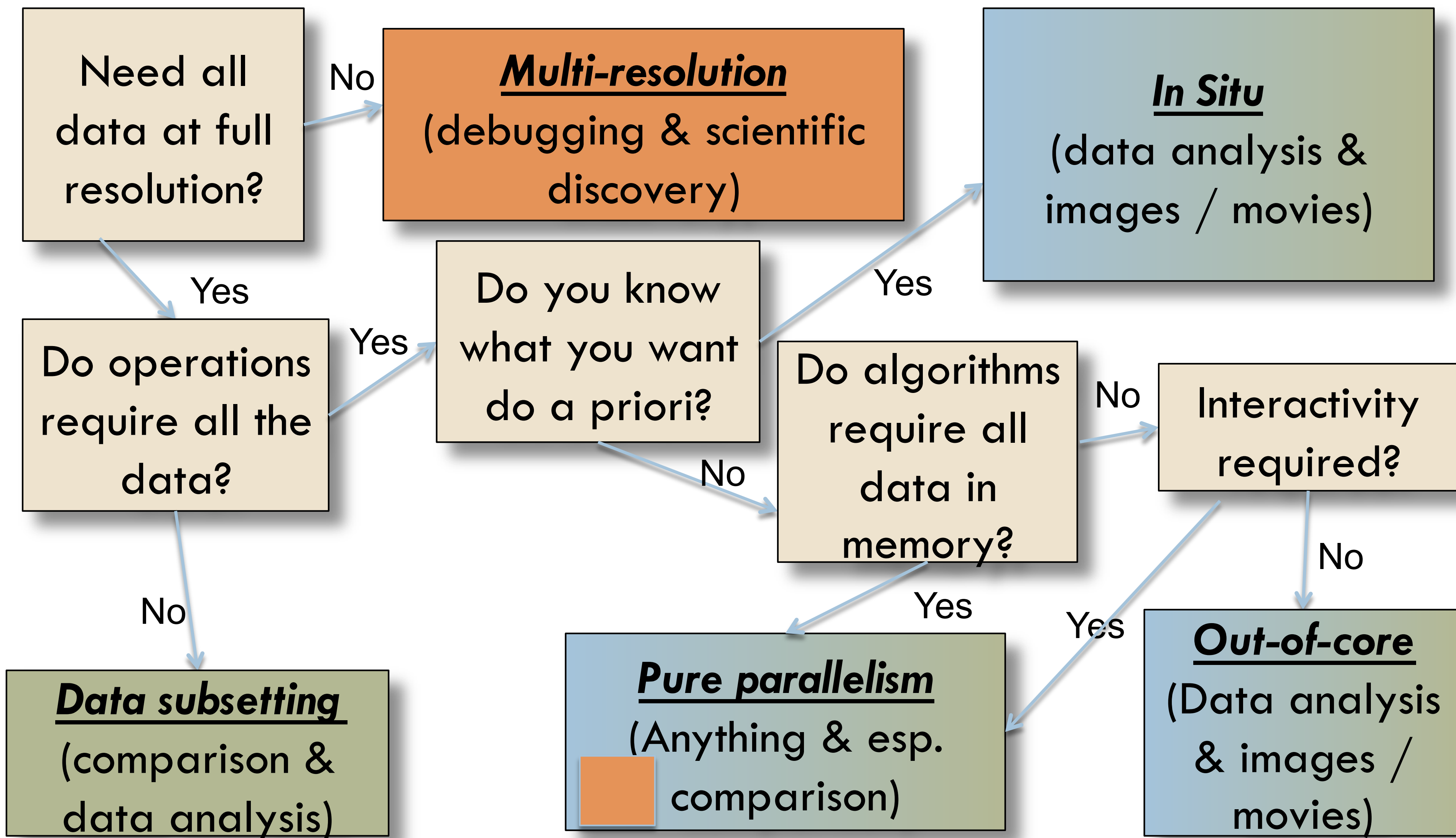
<u>Examples:</u>
Scientific discovery
Debugging

<u>Examples:</u>
Data analysis
Images / movies
Comparison

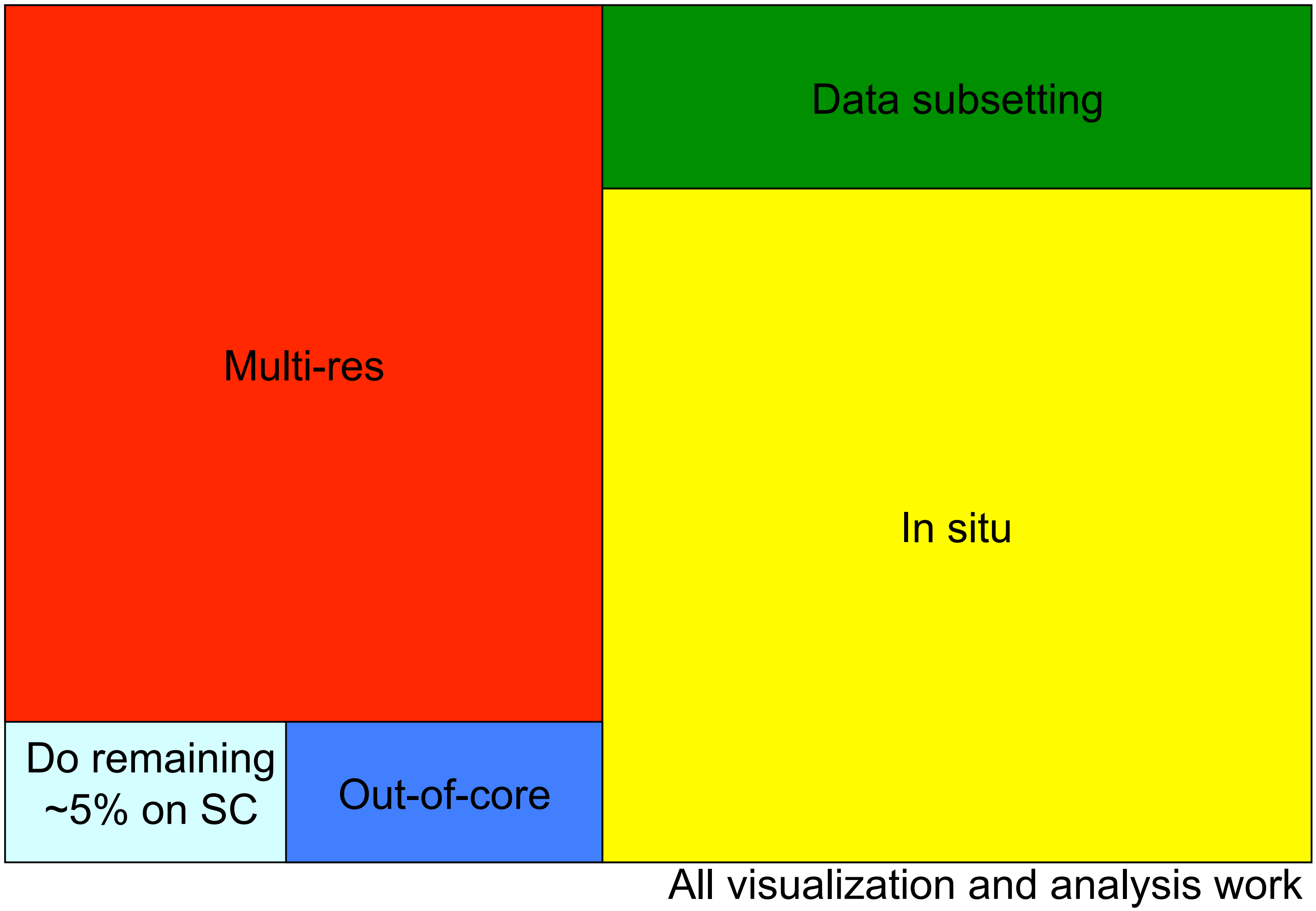
<u>Examples:</u>
Data analysis
Images / movies

Notional decision process

Exploration
Confirmation
Communication



Alternate strategy: smart techniques



Difficult conversations in the future…

- Multi-resolution:
  - Do you understand what a multi-resolution hierarchy should look like for your data?
  - Who do you trust to generate it?
  - Are you comfortable with your I/O routines generating these hierarchies while they write?
  - How much overhead are you willing to tolerate on your dumps? 33+%?
  - Willing to accept that your visualizations are not the "real" data?

Difficult conversations in the future…

- In situ:
  - How much memory are you willing to give up for visualization?
  - Will you be angry if the vis algorithms crash?
  - Do you know what you want to generate a priori?
    - Can you re-run simulations if necessary?

How Supercomputing Trends Will Changes the Rules For Vis & Analysis

- Future machines will not be well suited for pure parallelism, because of its high I/O and memory costs.
- We won’t be able to use pure parallelism alone any more
- We will need algorithms to work in multiple processing paradigms